Analysis of Search Algorithms on a Rush Hour Game

COMP 472 Assignment #2

Sarah Joyal

ID: 40083642

Length of Solutions

- For most random boards, the path is identical across search algorithms and heuristics used
- However, for some, the search path is longer for GBFS using heuristics h1, h2, and h3.
- For Example:

Board #	Algorithm	Heuristic	Moves
10	UCS	NA	16
10	GBFS	h1	18
10	A/A*	h1	16
10	GBFS	h2	18
10	A/A*	h2	16
10	GBFS	h3	18
10	A/A*	h3	16
10	GBFS	h4	16
10	A/A*	h4	16

+2 moves needed to find the solution when GBFS was used with certain heuristics, same does not apply for heuristic h4 (distance of AA to exit).

- Why is GBFS with heuristics h1, h2, and h3 not finding the same path as A/A* and UCS?
- GBFS only considers the heuristics, not the length of the path leading to the opened puzzle.

UCS	GBFS	A/A*
• Uses the total length of the path to the current node as priority of a node. • Guarantees to find the lowest-cost solution	• Uses only the heuristic of the opened node to calculate its’ priority. • Does not guarantee to find the lowest-cost solution, but may be faster than UCS and A/A*	• Uses both the total length of the path and the heuristic of the current node as its’ priority. • Guarantees to find the lowest-cost solution, but is time-consuming

- UCS and A* are both guaranteed to find the lowest-cost solution.

Heuristic Admissibility and Influence

h1(n)	h2(n)	h3(n)	h4(n)
Blocking vehicles between A and exit	Blocked spaces between A and exit	h1(n)*λ	Total spaces between A and exit
Admissible	Admissible	Not Admissible (unless $0 < \lambda \leq 1$)	Admissible

- Admissible heuristics never overestimate the cost of a solution
- Heuristic formula h3(n) is the only non-admissible heuristic because it is a multiple of the admissible heuristic h1(n). It can, however, be admissible if:

$$0 < \lambda <= 1$$

- Heuristic h4(n) is more optimal than h1(n), h2(n), and h3(n) when used with GBFS because it prioritizes the least number of moves it would take for the ambulance car to reach the exit, rather than focusing on moving other cars.

Execution Time

- The search algorithm with consistently faster results is GBFS, since its’ focus is execution time optimization.

Board	Algorithm	Heuristic	Moves	Search	Runtime
22	UCS	NA	36	3803	1.26
22	GBFS	h1	36	1677	0.58
22	A/A*	h1	36	3736	1.75
22	GBFS	h2	36	1677	0.57
22	A/A*	h2	36	3736	1.57
22	GBFS	h3	36	1677	0.62
22	A/A*	h3	36	3281	2.33
22	GBFS	h4	37	1915	0.69
22	A/A*	h4	36	3558	1.43

Despite these results, informed search is NOT always faster than uninformed search.

If a heuristic is non-admissible and unrelated to the solution of a problem, it can run infinitely or even run through every other path before finding the solution.

Since uninformed does not use a heuristic, it is unbiased and will eventually find a solution (or not if impossible), regardless of the puzzle.

Other Facts

- UCS and A/A* consistently return lower or equal solution paths to GBFS, but always take longer to execute and have larger search paths.

- Lengths of the search paths for GBFS on a board are identical between heuristics h2(n) and h3(n), but they change for A/A* on the same board.
  - ⇒ Since A/A* uses both path length and a heuristic, the path length will have less of an effect on a nodes’ priority if the heuristic is very large.

- Execution time of A/A* using h3(n) is drastically increased on certain puzzles, but its’ search path does not reflect the time increase.